COOLEY GODWARD LLP
ATTORNEY DOCKET NO.: CNTW-019/01US
CLIENT NO.: 036958-2041

EXPRESS MAIL NO. EL 955495288 US

APPLICATION FOR PATENT

TITLE: **REPOSITORY-INDEPENDENT SYSTEM AND METHOD FOR ASSET MANAGEMENT AND RECONCILIATION**

PRIORITY

[0001] This application claims priority to U.S. provisional patent application no. 60/395,698, entitled *Repository-Independent System and Method for Asset Management and Reconciliation*, which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The present invention relates to network device management. In particular, but not by way of limitation, the present invention relates to systems and methods for maintaining network device configurations and/or generating network device configurations.

BACKGROUND OF THE INVENTION

[0003] Network devices such as routers, switches and optical devices are becoming increasingly more complicated. Typical network devices now require thousands of lines of specialized configuration instructions to operate properly. Unlike most software applications, the instructions that operate network devices can be changed on a frequent basis, and the nature of network devices often requires that each version of a device's configuration be stored. Because changes are so frequent, sizable repositories of old configurations are generated for each device. When these sizable repositories are accumulated across the thousands of network devices that frequently make up a network, cumbersome, inefficient repositories are created. In some cases, these repositories are so large that they are not useful.

[0004] Present network architecture generally requires that configuration instructions and the capabilities of a network device (referred to as "configuration knowledge") be stored together as an atomic unit. This single-data-model approach has proven difficult to maintain for sophisticated networks. When network administrators, for example, archive only the configuration data—the actual configuration instructions or some indication thereof—the configuration knowledge that was used to generate those configuration instructions is lost. When the network administrators attempt to archive both the configuration instructions and the configuration knowledge for each configuration change, the size of the archived file becomes too large because the knowledge used to generate the configuration is many times the size of the actual configuration.

[0005] For a given version of a network device, the configuration knowledge is generally invariant, *e.g.*, the operating system and hardware for the network device do not change. Thus, repeatedly archiving the configuration knowledge is wasteful.

[0006] Network administrators have also found that the single-data-model implementation makes reverting to previous configurations difficult. When the configuration data and the configuration knowledge are bundled together as an atomic unit, network administrators have significant difficulty in reverting to a previous device configuration when both the configuration instructions and the configuration knowledge change. For example, when a network device is upgraded to run a new version of its operating system, both the configuration knowledge and the configuration data are

changed. If the upgrade fails, rolling back the changes to a known state for the previous operating system.

[0007] Present network technology suffers from yet another drawback in that it lacks a common information model that can be used to derive each of the application-specific configurations. This lack results in network applications having difficulty in retrieving and sharing network information from different network devices. Even more problematic is the fact that the lack of the common information model results in network applications sharing network data infrequently. For example, each application might implement its own procedure for discovery of network devices because it cannot understand information generated by another network application.

SUMMARY OF THE INVENTION

[0008] Exemplary embodiments of the present invention that are shown in the drawings are summarized below. These and other embodiments are more fully described in the Detailed Description section. It is to be understood, however, that there is no intention to limit the invention to the forms described in this Summary of the Invention or in the Detailed Description. One skilled in the art can recognize that there are numerous modifications, equivalents and alternative constructions that fall within the spirit and scope of the invention as expressed in the claims.

[0009] In one embodiment of the present invention, the configuration of a network device—also referred to as network resources—is separated into two portions:

configuration knowledge and configuration data. Configuration knowledge for a particular network device is referred to as a configuration knowledge instance. Similarly, configuration data for a particular network device is referred to as a configuration data instance.

[0010] Configuration knowledge abstractly represents the capabilities of a network device, but not necessarily the actual configuration of that device. For example, the configuration knowledge for a router might indicate the types of traffic conditioning, chip organization, and routing protocols that are available to that router. Configuration knowledge can be comprised of individual configuration schemata, which define the individual portions that make up the complete configuration knowledge.

[0011] Because configuration knowledge for a device can be constructed from a set of individual schemata, when the capabilities of that network device are changed, the relevant portion of the configuration knowledge instance can be changed without otherwise rebuilding the entire configuration knowledge instance. For example, if a new card is added to a router, then the schemata for that new card is added to the configuration knowledge instance. The remaining portion of the configuration knowledge instance, however, may remain unchanged.

[0012] The configuration data for a particular network device can be derived from the configuration knowledge instance for that device. Moreover, each configuration data instance can be associated with a particular version of the configuration knowledge

instance. For example, if a router is updated with a new operating system (OS), a new version of the configuration knowledge instance that reflects the new OS is created. Subsequent sets of configuration data can be associated with the new version of the configuration knowledge instance.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Various objects and advantages and a more complete understanding of the present invention are apparent and more readily appreciated by reference to the following Detailed Description and to the appended claims when taken in conjunction with the accompanying Drawings wherein:

FIGURE 1 illustrates one organization of a configuration knowledge instance for a network device;

FIGURE 2 is a block diagram of one embodiment of the present invention;

FIGURE 3 illustrates versioned KDMs and configuration instructions;

FIGURE 4 is a block diagram of a network including network management applications and configuration knowledge and data storage devices;

FIGURE 5 is a flowchart of one method for implementing a roll-back; and

FIGURE 6 is a flowchart of one method for implementing a business policy in a network.

DETAILED DESCRIPTION

[0014] Individual network devices are typically associated with a device configuration that controls the operation of that network device. In one embodiment of the present

invention, the device configuration for network device is separated into two portions: configuration knowledge and configuration data. Configuration knowledge abstractly represents the capabilities—both logical and physical—of a network device, and the configuration data includes information about the actual configuration of the network device. Put simply, configuration knowledge describes the features of a network device, and the configuration data indicates which features are being used and how they are being used.

[0015] Typical configuration knowledge can include separate abstractions for each feature of the network device. For example, the configuration knowledge for a particular router could list the physical properties of the router such as processor type and available cards. Similarly, the configuration knowledge could list the logical capabilities of the router such as available protocols, security features and services. The actual configuration information for these physical and logical properties would be stored with the configuration data instance for that router. Note that the configuration of most network resources, including routers, router components, switches, switch components, fabrics, optical devices, and optical components can be divided into configuration knowledge and configuration data.

[0016] Referring now to FIGURE 1, it illustrates one possible organization 10 of configuration knowledge. This abstraction includes a device family layer 12 for devices that all share common features and/or other characteristics. A typical device family could be "router" or "CISCO router." The device family layer 12 is refined by the device layer

14, which represents a software abstraction of a specific device. A typical device for the "router" family could be "CISCO," and a device for the "CISCO router" family could be a particular model of CISCO router. The device family layer 12 can then further refined into its physical and logical aspects, which are represented by the physical and logical abstraction layers 16 and 18.

[0017] The physical and logical layers 16 and 18 can be refined according to the features of the family of devices being represented. For example, the logical abstraction for a router can include: address management, services, security, protocols, and traffic conditioning. Similarly, the physical abstraction can include: cabling, processors, cards, and chassis. These refinements are not inclusive, but rather exemplary for one type of device. Note that the logical and physical layers represent the capabilities of the class of network devices and not the actual configuration of any particular device.

[0018] By defining the device according to its physical and logical capabilities, configuration knowledge can support applications that require access to only physical or logical information. For example, configuration knowledge can be used to support a physical inventory application that has no need of logical information. Likewise, the configuration knowledge can support a capacity planning application that has need for both physical and logical information. In either case, the application seeking information need only query the configuration knowledge and not the actual configuration as stored in the configuration data.

[0019] Configuration knowledge can be organized using object classes, directories, and inheritance properties. For example, the template for a new instance of configuration knowledge for a CISCO router could be formed by creating an instance that inherits the properties of a "CISCO router" class, which inherits the properties of the generic "router" class. The template would then be populated with the specific information, such as available cards and operating systems, pertaining to the particular CISCO router being modeled.

[0020] Once created, individual instances of configuration knowledge can be stored together in a central storage device or distributed across multiple storage devices. For example, the configuration knowledge instances for each router on a network could be stored together in a central facility. The configuration knowledge instances can be stored in a variety of formats, including XML.

[0021] Referring now to FIGURE 2, it is a block diagram of one embodiment of the present invention. In this embodiment, instances of configuration knowledge and configuration data are stored in a configuration storage device 20. The configuration storage device 20 is represented as a single device for simplicity only. It could be arranged in any fashion, including distributed, centralized, or some combination thereof. Additionally, a particular configuration knowledge instance and configuration data instance could also be stored at the network device 24 to which they correspond.

[0022] The configuration storage device 20 is connected to a management application 22 that can be implemented in software or hardware. Additionally, the management

application 22 can consist of several individual applications, including applications distributed over a network. The management application can be responsible for several functions, including:

- **Facilitation of search and accounting of assets**

The management application 22 can search the individual configuration knowledge instances for particular capabilities. For example, the management application 22 can search for device capabilities such as hardware and software features of a network device that are no longer being used and are otherwise available. For example, consider the creation of a VPN. This requires dedicating either an interface or a sub-interface of a Physical Port of a network device to host the VPN traffic, along with dedicating logical resources that correspond to creating the instance of the VPN. This enables the network device to forward traffic on the VPN if the traffic is intended for that VPN. One example of a search is to identify components of a VPN. Similarly, if the VPN is subsequently removed, then it is important to reclaim these allocated resources. Thus, a second example of a search is to ensure that the components have been removed. A third example of a search is to ensure that adequate resources for creating the VPN exist before the commands are issued to the device. The management application 22 could also search the configuration knowledge instances for stranded services such as a virtual private network (VPN) that is no longer being used. Similarly, the management application 22 could search for software capabilities, physical ports, physical assets, and physical containers. In effect, the management application 22 can provide an accurate

inventory of the capabilities of a network. Such information can be used for network management, provisioning, and identification of stranded assets.

- **Support for versioning of asset information**

The management application 22 can also support versioning of both configuration knowledge and configuration data. For example, multiple versions of configuration data could be associated with a single instance of configuration knowledge. Such versioning is particularly useful for creating different instances of configuration data that can be associated with different customer demands. Versioning is described in more detail with relation to Figure 3.

- **Support for concurrent editing of asset information**

The management application 22 can also enable different users to work on different parts of the configuration knowledge and configuration data simultaneously.

- **Support for incremental update to versioned asset information**

The management application 22 can also track which individual features of a network device are changed and how those changes impact the configuration data. For example, if an updated card was added to a particular router, then the management application could change only the portion of the configuration knowledge corresponding to the updated card. Similarly, only the portion of the configuration instructions corresponding to the changed portion of the configuration knowledge need be changed.

[0023] Referring now to FIGURE 3, it illustrates a versioned configuration knowledge instance and corresponding versions of configuration data. In this embodiment, the configuration knowledge instance is associated with a particular network device and

includes versions 1 through 4. The configuration data also corresponds to the network device and includes versions 1.1 through 4.3. Each version of the configuration data instance is associated with at least one of the versions of the configuration knowledge. For example, configuration data V1.1 and V1.2 correspond to configuration knowledge instance V1. Similarly, configuration data instance V2.1 corresponds to configuration knowledge instance V2.

[0024] Referring now to FIGURE 4, it is a block diagram of a system that includes network management applications 40 connected to a centralized configuration knowledge storage device 42 and configuration data storage device 44. In this embodiment, a plurality of network management applications 40 are connected to the storage device through a network 46. The storage devices 42 and 44 are also connected to network devices 48(a) and (b) such as router through the network.

[0025] When a network management application 40 needs configuration data about a particular network device or group of network devices, the network management application 40 can access the network device 48 directly and read the relevant information. This process, however, generally requires the network management applications 40 to understand the particular syntax of the network device's configuration. In one embodiment of the present invention, however, the network management application 40 can access the storage device 42 or 44 and retrieve the relevant configuration knowledge instances or portions thereof.

[0026] Because the configuration knowledge instances are abstractions of the capabilities of the device, the network management applications 40 generally are not required to understand the device-specific syntax of a particular network device. For example, a physical inventory application could access the configuration knowledge instances for the relevant network devices and determine the cards that are used by each device without regard to the syntax of the actual configuration instructions.

[0027] Referring now to Figure 5, it is a flowchart of one method for implementing a roll-back using configuration knowledge instances and versioned configuration data. Roll-backs are often useful for network administrators after network attacks or after unsuccessful network device updates—although they are useful in several other cases. For example, new hardware is often added to existing routers in a network. This new hardware can introduce new capabilities to the router that are reflected in a new version of the router's configuration knowledge instance. Additionally, the configuration data for the router is usually modified to engage the new hardware. Thus, in this type of system upgrade, both the configuration knowledge instance and the configuration data instance for the router are modified.

[0028] Assuming that a system upgrade is unsuccessful for some reason, network administrators often wish to roll-back the configuration to a previous, known configuration. For example, if the added card was defective, the network administrator might want to remove the defective card and roll-back the configuration to a configuration based on router that does not include the card. To roll-back the

configuration, the assembler or some other device can identify the device [step 50] and a version of the configuration knowledge instance that does not reflect the card's presence [step 52]. The configuration data associated with that version of the configuration knowledge instance can then be identified [step 54] and pushed to the network device [step 56].

[0029] Referring now to FIGURE 6, it is a flowchart of one method for generating a business object model (BOM) for implementing a specific purpose in a network. In this embodiment, a user or application requests a device configuration to perform a certain function [step 60]. For example, assume that it is desired to create a VPN. The actual list of commands required to accomplish this task vary by vendor and also by version of the operating system that the network device is running. Therefore, in order to provide a single high-level ability to create a VPN, detailed knowledge of the differences in command syntax and semantics must be provided. In various embodiments of this invention, this is done through the use of a BOM, which correlates and assembles the individual knowledge instances. In a preferred embodiment of this invention, there will be many such BOMs, with a BOM for each type of function. Note that the function can be small or large, a command change or a VPN creation being examples of each. For the VPN creation, there will be a set of BOMs that are aggregated into a higher-level BOM. This request is handled by a BOM assembler. The BOM assembler determines which network resources are required to carry out the request [step 62]. The BOM assembler next gathers information from the configuration knowledge instances associated with the identified network resources [step 66].

[0030] Finally, the BOM derives the device configuration from the gathered configuration knowledge instances and generates the actual configuration commands [step 70]. For example, in the configuration of a VPN, the BOM assembler will select appropriate BOMs, aggregate them together, and use the aggregated BOM to derive the appropriate device configuration commands for each device. This enables the device to be programmed at a high functional level, and to have these high-level functions translated to a low-level device-specific implementation. Examples of systems for generating commands are described in commonly owned and assigned U.S. patent application nos. 09/730,671, entitled "Dynamic Configuration of Network Devices to Enable Data Transfers," and 09/730,864, entitled "System and Method for Configuration, Management, and Monitoring of Network Resources," both of which are incorporated herein by reference. In one embodiment, the device configuration is derived by binding the variable information within the configuration knowledge instances to the business purpose of the customer. For example, a QoS business purpose could be bound to the various traffic conditioning settings.

[0031] In conclusion, the present invention provides, among other things, a system and method for managing and utilizing network device configurations. Those skilled in the art can readily recognize that numerous variations and substitutions may be made in the invention, its use and its configuration to achieve substantially the same results as achieved by the embodiments described herein. Accordingly, there is no intention to limit the invention to the disclosed exemplary forms. Many variations, modifications and

alternative constructions fall within the scope and spirit of the disclosed invention as expressed in the claims.